The name of my project: Monopoly - Be the last player remaining

- Description:

In this game, you could either win by buying and developing your own property and forcing your rivals into bankruptcy, or you could lose by experiencing your own bankruptcy. Being the last player remaining with money is your goal.

- Similar projects:

I've seen one person making an AI in her monopoly game. She utilized a one-dimensional graph. In her game, she set a timer to control the time length of the game at 1 hour (1200s). AI will choose whether or not to buy the property of the land in the current round, and its money will aggregate as the game continues. If there is a negative growth of one's assets, player and AI could press S to sell his/her/their property to pay the debt. The fundamental rules of monopoly in my game would be similar to those in hers. But I would not add any AI into it; instead, I'd make it a multiplayer game. After MVP, I might try using a socket to accomplish it if at all possible.

In another person's TP, I saw him using isometric game graphics, and this is quite close to what I have in mind. Apart from that, I would like to add some function cards for players to use.

- Structural Plan:

I would first create a class Player, so my game would allow multiplayers to play and the actions in the Class Player would apply to every instance of the players.

Then I would utilize a 2-D list (I might somehow use random function here to create the list), consisted of True and False, indicating the road of the game and finally I would build the map (the monopoly world) based on the list.

- Algorithmic Plan:

The most challenging step would be to create an isometric graphical map. My current plan is to construct a function that takes in the app and the coordinates of the center point of the one-dimensional grid that I want to create, calculates the size of the one grid proportioned to the size of the canvas and returns a 2.5-dimentional grid.

Apart from this, I want to create several maps for the players to choose from in my game. So I'm wondering if I should hard code the shapes of the maps, or to use some function to randomly create a map.

Another difficulty might be to move the player from one grid to another, because it would be built on an isometric game graphics, so the route would be a little skewed. But right now I don't think it's conceptually impossible. I could calculate the horizontal and vertical distance that the player would move in each direction in order to move them.

Timeline Plan:

to-do list:

1. initialize the basic setups of a player, create a Class of Player

- In the player Class, define different moves that the player could take, including

- 1)play a dice/ change current grid that the player is on; 2)buy a property/ upgrade a property; 3) check the player's current state (qualified to move or not); 4) use a function card (maybe)

2. create a map in the representation of a 2-D list

- make each grid on the board a dictionary

- initialize the basic setups of a board, including

- 1)the name of each place in the map(maybe); 2)the price to buy each place(use random.randint); 3)the price to upgrade the place to each level; 4)the price of the property if the owner decide to sell it (or if the owner becomes bankrupt and has to sell his/her/their property to pay the debt); 5) the price that other players need to pay when visiting that grid

3. allow players to move on the board, check next movement-> make movement

4. set up function cards and their corresponding change in the player instance

5. write keyPressed() to allow players to take actions

6. (highlight, difficult) draw the map on canvas

7. make the map more aesthetically appealing

8. write instructions

By the end of Wed (11.16) - finish step 1 and get started on step 2

By the end of Sat (11.18) - finish step 2 and step 3

11.20 TP1 due

By the end of Wed (11.23) - get started on step 6 and finish step 4 and 5

By the end of Mon (11.28) - finish step 6

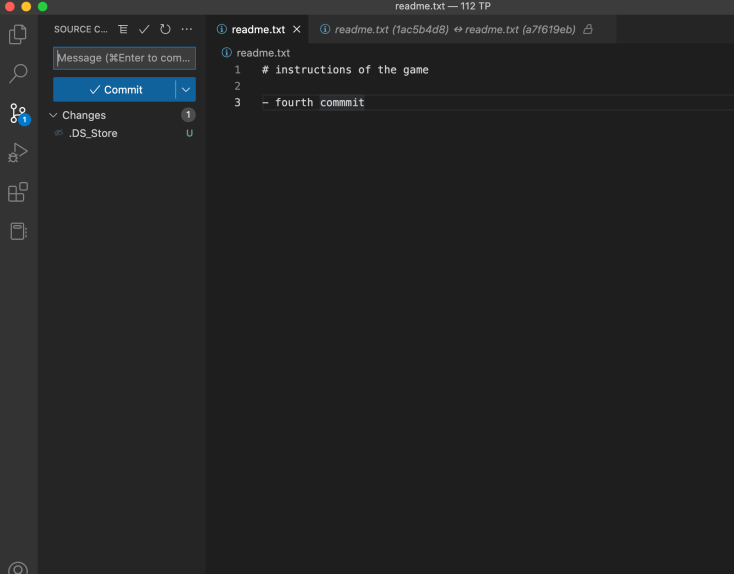
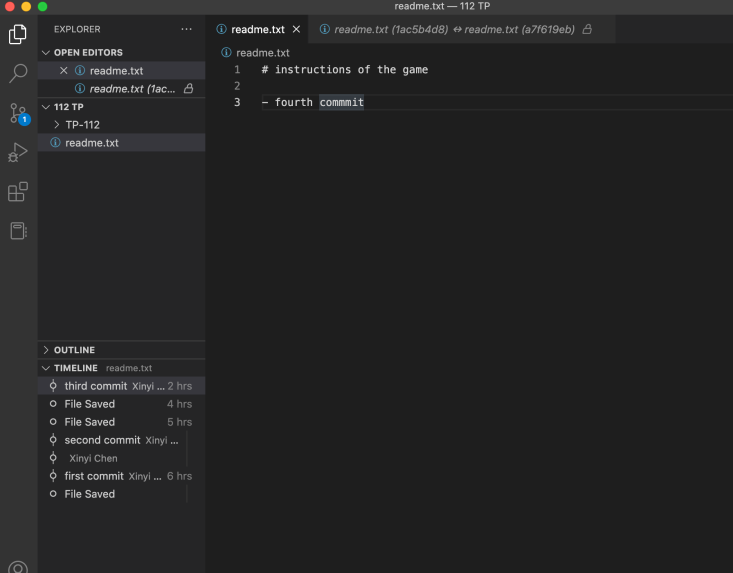
11.30 TP2 due

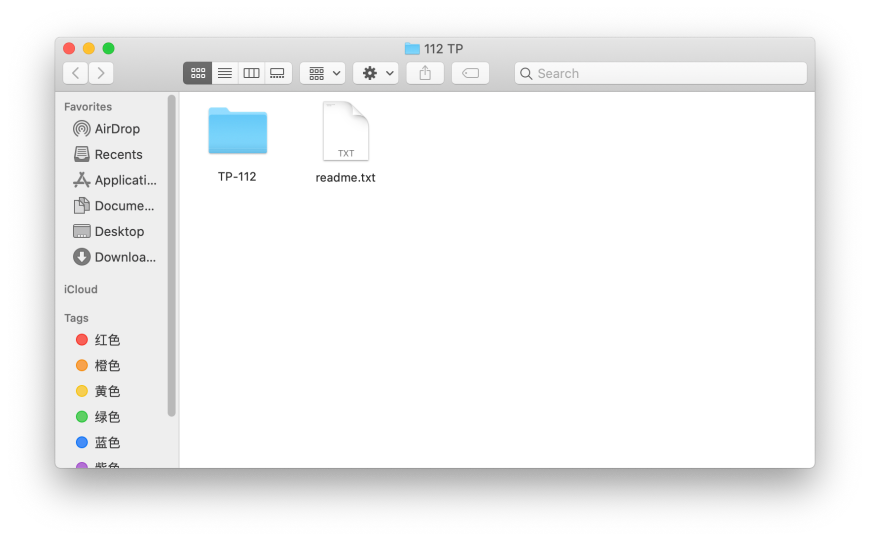
By the end of Sun (12.4) - finish step 7 and 8

Version Control Plan

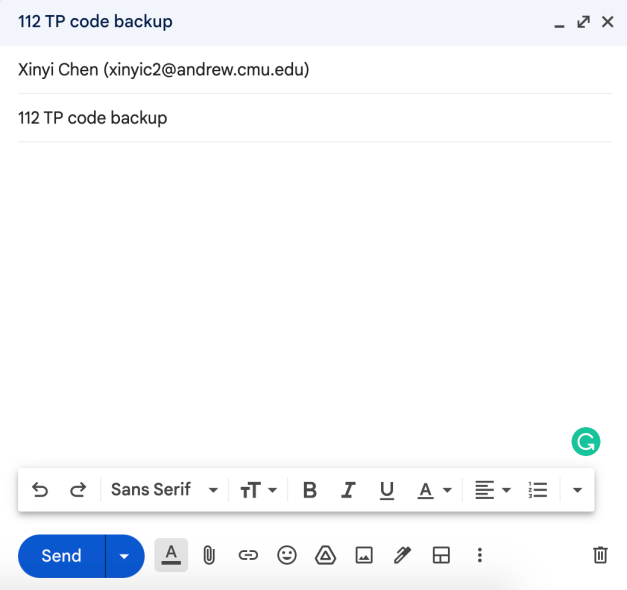
- I have two plans in mind: first I will use Git to back up my code; then I will send the code to my email address at the end the day every day.

The below screenshot should be my repository, but I'm not very sure.





And this is how I will send the code to myself.



Module List:

I will not be using any external modules.

**TP1 Update**

I will be using Github to back up my code.